

AMENDMENTS TO THE CLAIMS

1.-20. (Cancelled)

21. (Currently Amended) A method in a computer system for creating a file system snapshot, the data of the file system being organized hierarchically via nodes, the method comprising:

associating with one or more current file system nodes an indication that said nodes are not to be part of said a subsequently created snapshot;

copying a root node of the file system to a new node that points to the same child nodes of the root node, wherein the new node represents a root node of the snapshot; and

when a node of the file system is modified,

checking the node and ancestor nodes of the node for said indication; and

~~replacing said node and said ancestor nodes of the node that have not yet been replaced with one or more new nodes only if said indication is absent~~

if said indication is found, performing the modification without replacing any nodes.

22. (New) A method in a computer system for creating a snapshot in a file system while modifying data associated with a file in the file system, wherein the file system is organized hierarchically via file and directory nodes, said file and directory nodes comprising current nodes and snapshot nodes after creation of the snapshot, the method comprising:

copying data from a current root node of the file system to a snapshot root node, wherein said snapshot root node has one or more current nodes descendent therefrom;

replacing current nodes with new nodes as current nodes of the file system are modified by file or directory changes; and

recording information for each current node indicating whether the node was replaced due to current node modifications since the snapshot root node was created;

wherein said replacing comprises:

(a) using said recorded information to determine if a particular current node to be modified has been previously replaced with a new node since the snapshot root node was created;

(b) if said particular current node to be modified has not been previously replaced with a new node, using said recorded information to determine whether one or more ancestor current nodes of said particular current node have not been previously replaced with a new node since the snapshot was created;

(c) if one or more ancestor current nodes have not been previously replaced, copying data from those current ancestor nodes that have not been previously replaced to one or more new current ancestor nodes that are descended from said current root node but not descended from said snapshot root node; and

(d) if said particular current node to be modified has not been replaced, copying data from said particular current node to be modified to a new current node descended from said new ancestor node but not descended from said snapshot root node.

23. (New) The method of Claim 22, wherein said particular node is a file node and said ancestor node is a directory node.

23. (New) The method of claim 22 wherein each new node created in said copying steps is associated with a snapshot identifier that identifies the snapshot during which it replaced a node and including checking the snapshot identifier of an ancestor node to determine whether it has been replaced during the current snapshot.

24. (New) The method of claim 24 wherein when a node is not to be part of a snapshot, associating an indication with that node so that node will not be replaced when it or any descendant node is modified.

25. (New) The method of claim 22 wherein the snapshot is accessed via the root node of the snapshot.

26. (New) The method of claim 22 wherein each new current node has an identifier that is different from the identifier of the node it replaced.

27. (New) The method of claim 26 including associating the identifier of each new current node with the identifier of a snapshot node so that, when a request to access a new current node originally identified by the identifier of a snapshot node is received, that association is used to access the new current node.

Appl. No. : 10/616,411
Filed : July 8, 2003

28. (New) The method of claim 27 wherein the associating includes storing the identifier of the new node in the snapshot node.

29. (New) The method of claim 27 including associating the identifier of the replaced node with the replacing node so that, when the identifier of the replacing node is requested, that association is used to provide the identifier of the replaced node.

30. (New) The method of claim 22 wherein each node has a reference count that includes a count of the snapshots through which the node is accessible.

31. (New) The method of claim 22 wherein the file system is a Unix-based file system.

32. (New) The method of claim 22 wherein a snapshot identifier is stored within at least some nodes.

33. (New) The method of claim 22 wherein a snapshot identifier is stored as an attribute of at least some nodes.

34. (New) The method of claim 22 wherein a virtual identifier is stored within at least some nodes.

35. (New) The method of claim 22 wherein a virtual identifier is stored as an attribute of at least some nodes.

36. (New) The method of claim 22 wherein when a file is modified, the new node associated with that file is set to reference the data for the modified file, rather than the data for the unmodified file.

37. (New) The method of claim 22 wherein when a block of a file is modified, the new node associated with that file is set to reference a block that contains the modified block, rather than the block that contains the unmodified data.

38. (New) The method of claim 37 including reference counting each snapshot that refers to a block so that the block can be removed when there are no more references to the block.

39. (New) The method of claim 38 wherein the reference counting is performed using a table external to the block.

Appl. No. : **10/616,411**
Filed : **July 8, 2003**

40. (New) The method of claim 39 wherein the table includes for each block a bit for each snapshot that indicated whether the block is referenced by the snapshot.